

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 9

ВАРИАНТ 22

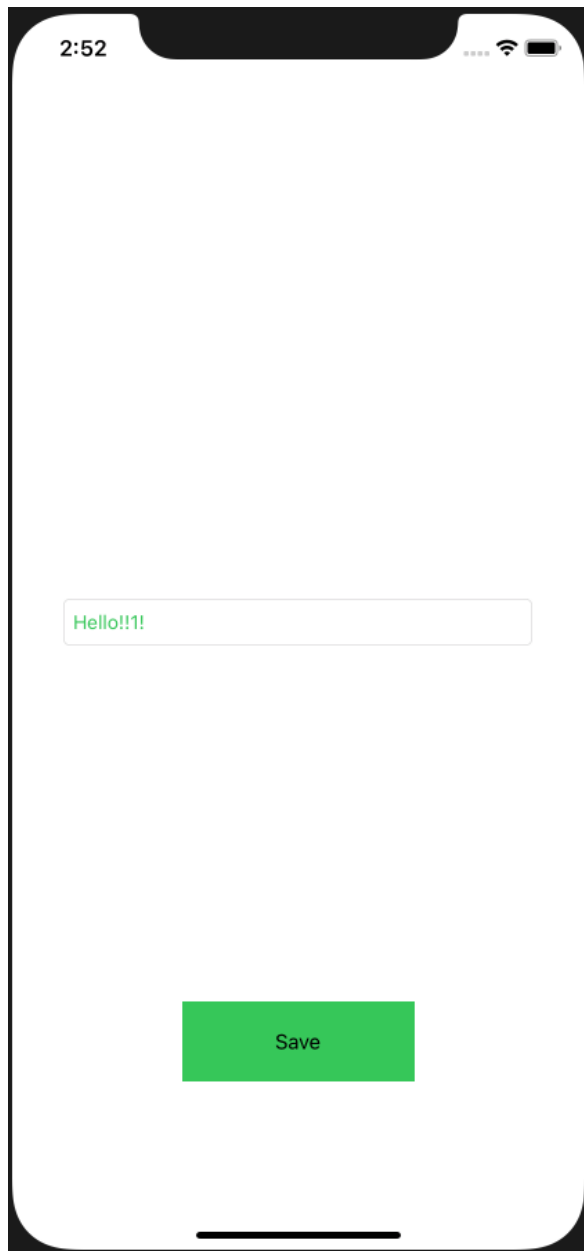
КРАЖЕВСКИЙ АЛЕКСЕЙ

1. ПРИМЕРЫ ДЛЯ ИЗУЧЕНИЯ

1.1. IOS ПРИЛОЖЕНИЕ С ХРАНЕНИЕМ ДАННЫХ В PLIST НА ЯЗЫКЕ SWIFT

Цель: изучить примеры использования plist.

Главный экран полученного приложения:



В примере было разработано приложение, которое берет данные из .plist файла и выводит на экран телефона. Кнопка ничего не делает). Сам файл myData.plist:

example1 > example1 > myData.plist > No Selection

Key	Type	Value
▼ Root	Dictionary (1 item)	
myitem	String	Hello!!1!

1.2. ПРИЛОЖЕНИЕ С АВТОРИЗАЦИЕЙ И ХРАНЕНИЕМ ДАННЫХ В NSUSERDEFAULTS НА ЯЗЫКЕ SWIFT

ЗАДАЧИ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Требуется разработать приложение с формой авторизации и хранением данных пользователей в UserDefaults

Входные данные

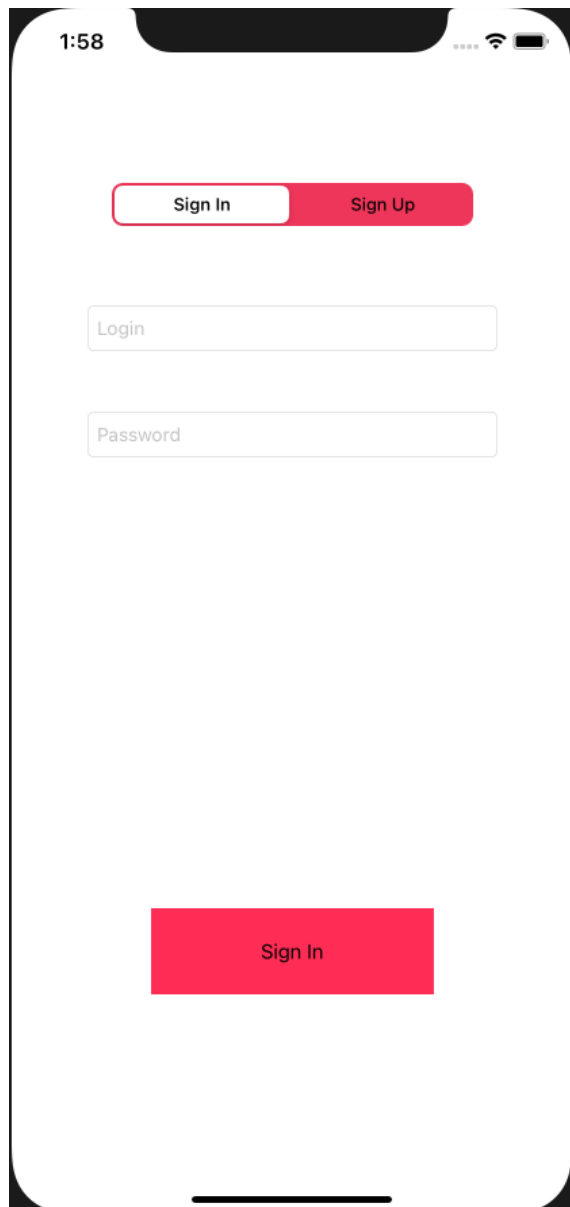
Форма авторизации / регистрации

Выходные данные

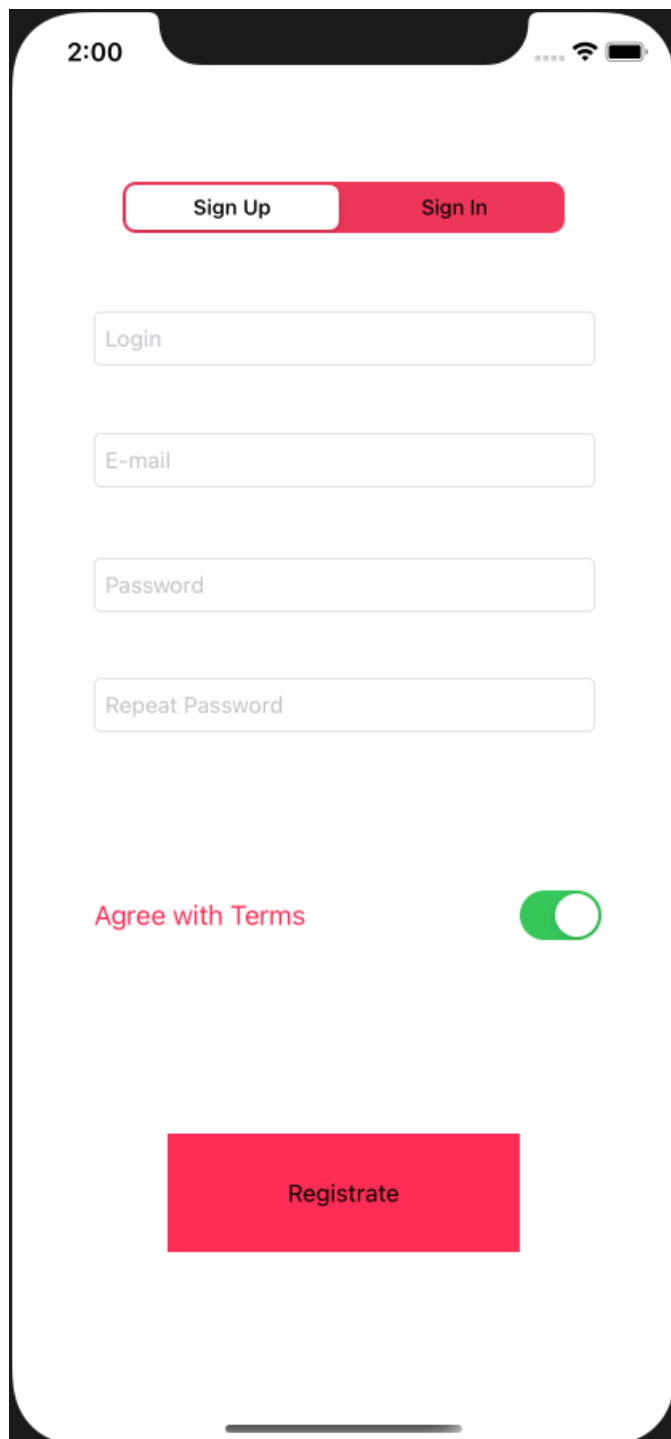
Уведомление об успешной авторизации или регистрации

Как я понял, приложение должно быть на objective-c.

Главный экран полученного приложения:



Посмотрим экран регистрации:

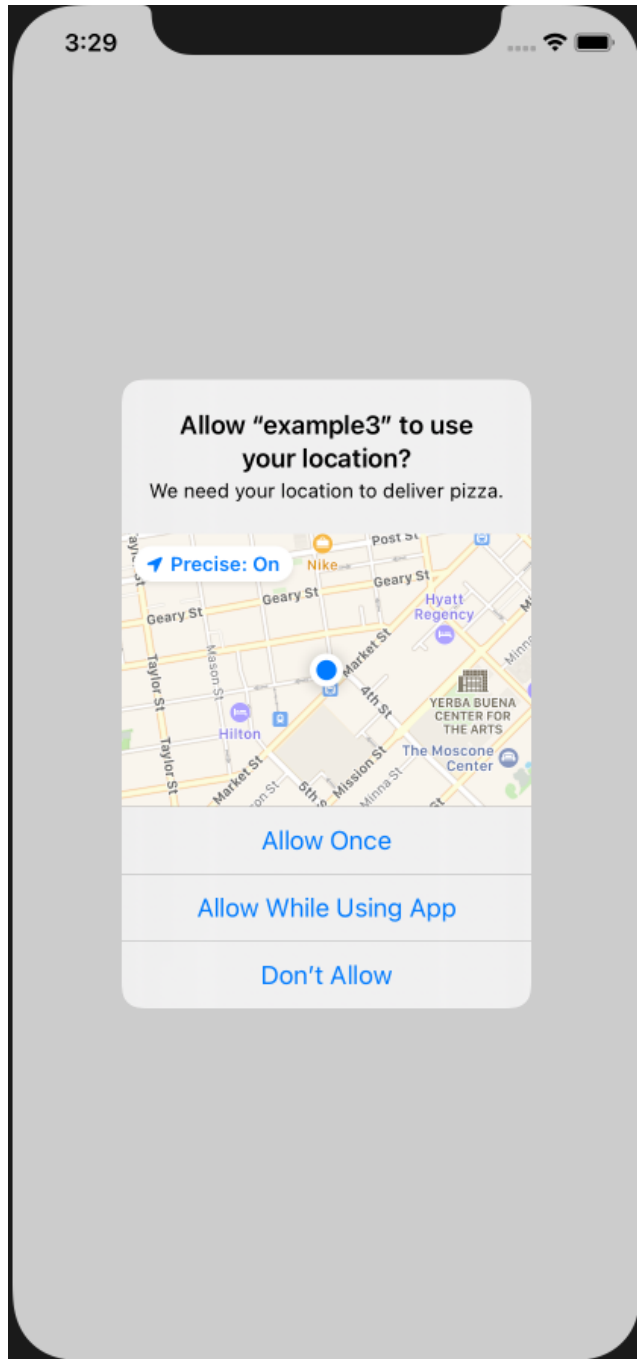
A mobile application registration screen. At the top, the status bar shows the time 2:00, signal strength, Wi-Fi, and battery. Below the status bar, there are two buttons: 'Sign Up' (white with a red border) and 'Sign In' (solid red). Under these buttons are four text input fields: 'Login', 'E-mail', 'Password', and 'Repeat Password'. Below the input fields, there is a red text label 'Agree with Terms' followed by a green toggle switch that is currently turned on. At the bottom of the form is a large red button labeled 'Registrate'. The entire form is enclosed in a black border.

Итак, если пароли не будут совпадать или если не будут заполнены все поля или если нет согласия с условиями, регистрация не пройдет. В ином случае пользователь будет успешно зарегистрирован и при следующем запуске приложения пользователь сохранится. При входе в аккаунт все вышесказанное остается верным.

1.3. IOS ПРИЛОЖЕНИЕ С ФУНКЦИЕЙ ГЕОЛОКАЦИИ НА ЯЗЫКЕ SWIFT

Цель: изучить примеры определения местоположения пользователя с использованием библиотеки CoreLocation

Главный экран полученного приложения:



При разрешении получения геолокации она выводится в консоль:

```
CLLocationCoordinate2D(latitude: 37.785834, longitude: -122.406417)
```

Для работы приложения были добавлены 2 Custom IOS Target Properties для получение разрешения для получения геолокации.

```
Privacy - Location Always and When In Use Usage... String We need your location to deliver pizza always.
```

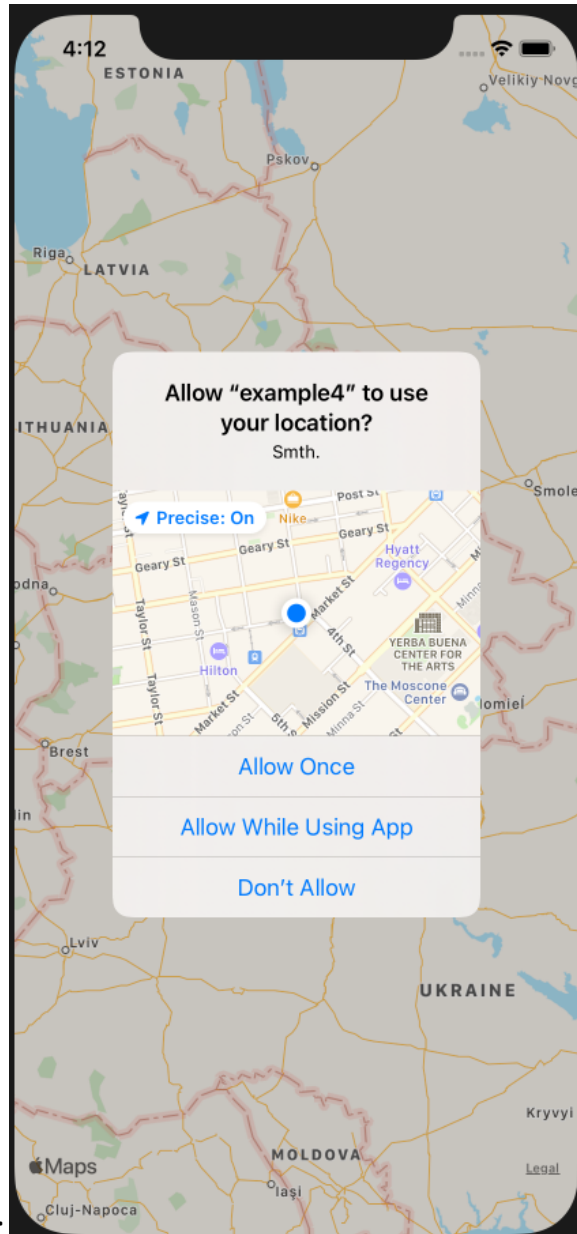
```
Privacy - Location When In Use Usage Description String We need your location to deliver pizza.
```

Вся остальная работа происходит в коде программы.

1.4. IOS ПРИЛОЖЕНИЕ С ФУНКЦИЕЙ ОПРЕДЕЛЕНИЯ МЕСТОПОЛОЖЕНИЯ ПО КАРТЕ НА ЯЗЫКЕ SWIFT

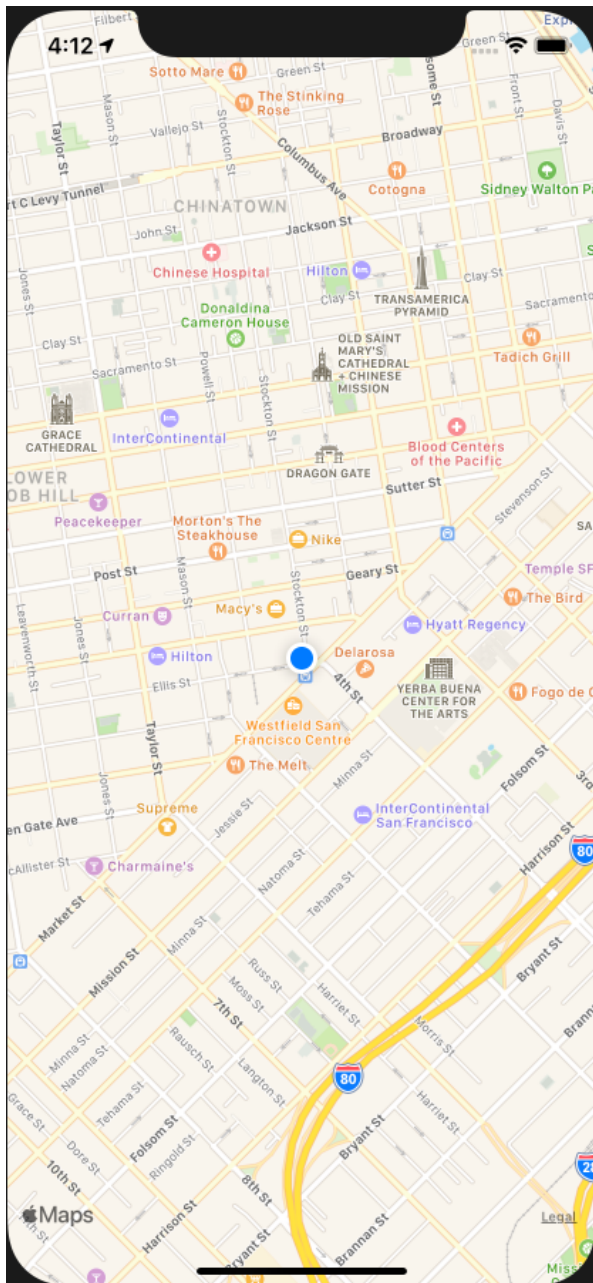
Цель: изучить примеры определения местоположения пользователя с использованием библиотек MapKit и CoreLocation

Экран после



запуска приложения:

После того, как пользователь дал разрешение:



Тут также были добавлены 2 Custom IOS Target Properties для получение разрешения для получения геолокации.
Вся карты выводится на MKMapView, сама геолокация берется из configpack, который создан для проверки корректности работы приложения.

Заглянем в БГУ:



1.5. СИСТЕМА БРОНИРОВАНИЯ БИЛЕТОВ НА ОБЪЕКТИВ-С ЗАДАЧИ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Изучить пример и разработать программу бронирования авиабилетов, используя CoreData и MapKit. Пользователь выбирает маршрут на карте, и ему предлагается список рейсов, включающих название авиакомпании, стоимость перелёта. Данные должны храниться в базе данных sqlite, доступ через CoreData.

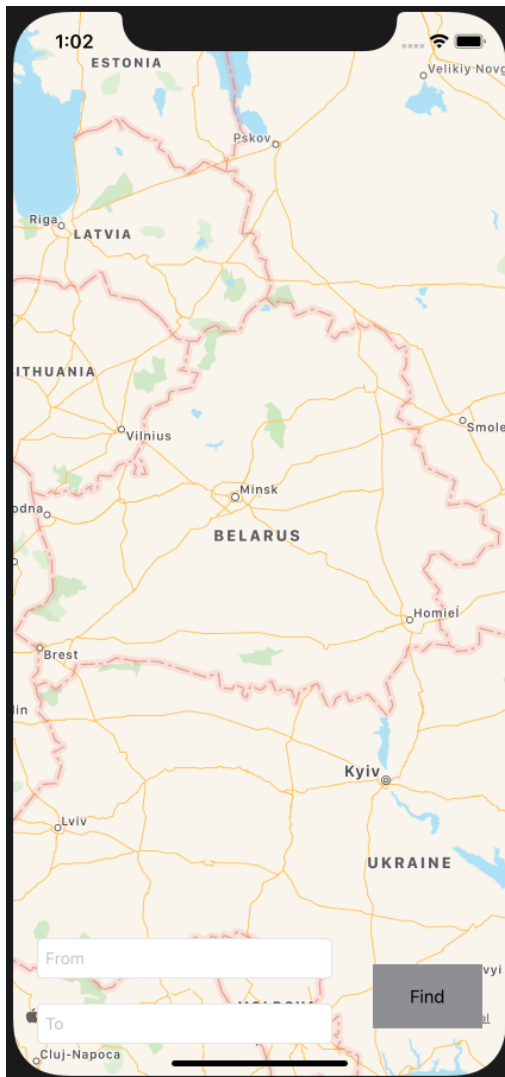
Входные данные

Маршрут, состоящий из набора городов.

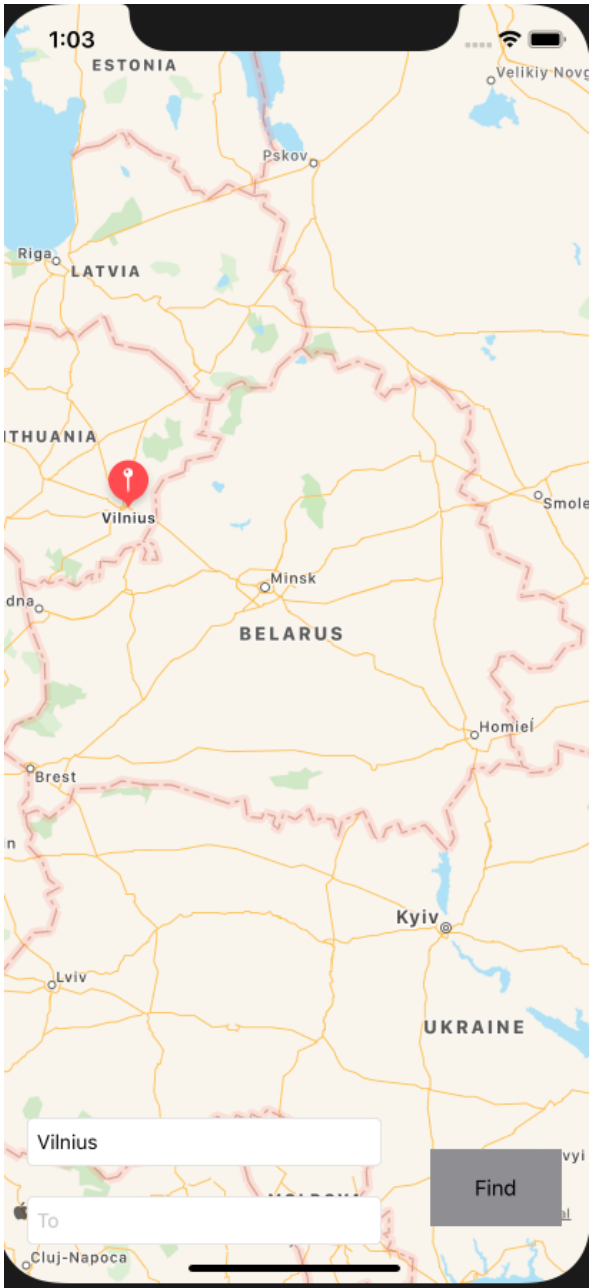
Выходные данные

Список рейсов, включая время и стоимость.

Главный экран полученного приложения:



Долгое нажатие на город:



1.6. IOS ПРИЛОЖЕНИЕ С ИСПОЛЬЗОВАНИЕМ БАЗЫ ДАННЫХ CORE DATA

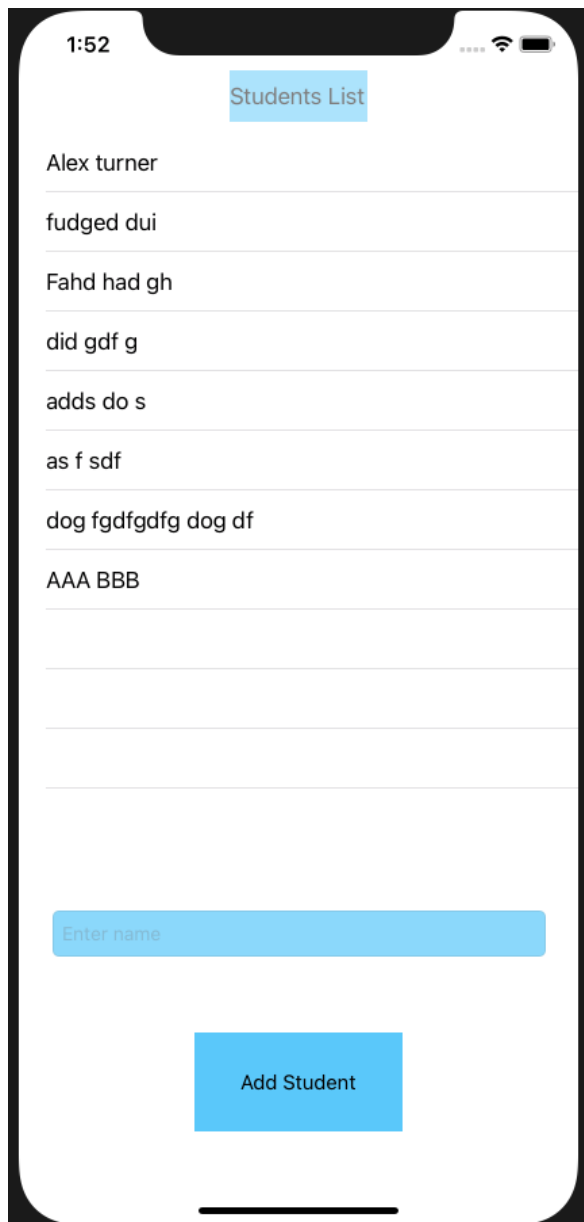
Цель работы:

Научиться работать со встроенными в Xcode средствами для работы с базами данных – Core Data. Научиться добавлять, удалять и получать объекты из базы данных.

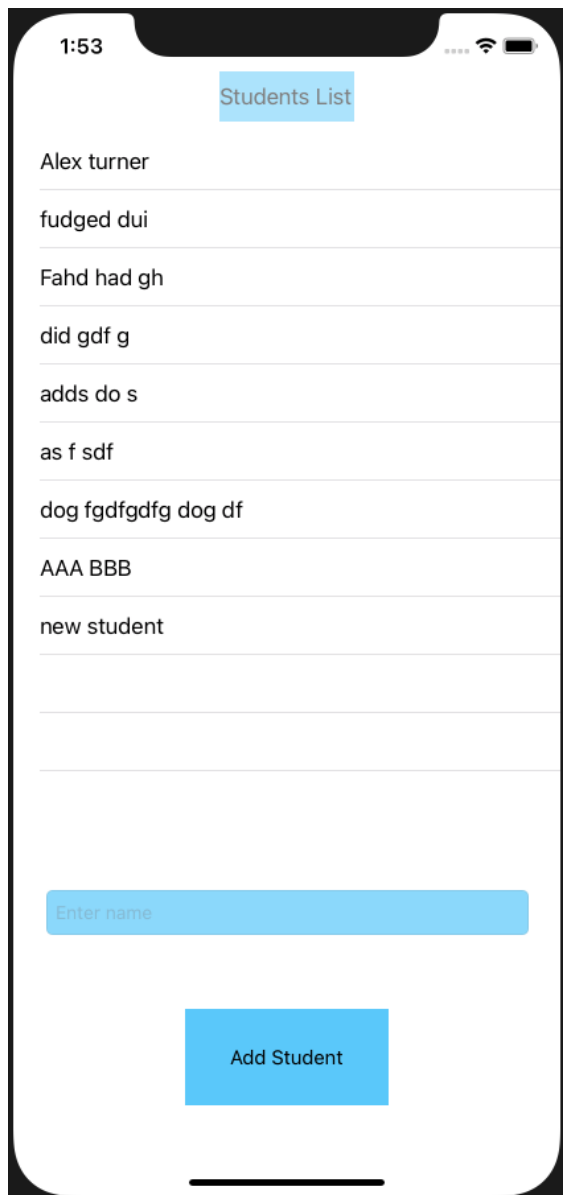
Указания к работе:

Изучить пример создания приложения, которое позволит хранить список студентов в базе данных. При закрытии приложения, данные должны сохраняться. Реализовать возможность добавления и удаления студентов из списка. Список должен быть отображен в объекте Table View. Добавление должно происходить с помощью объекта Text Field по кнопке Button. Перед выполнением просмотрите видео <https://www.youtube.com/watch?v=2TYvcdBPMB4> и в ходе работы внесите изменения в приведенный код в соответствии с синтаксисом языка Swift3.

Главный экран полученного приложения:



Кнопка Add Student добавляет новую запись в базу данных:



Сущность Students для хранения данных о студентах:

ENTITIES

Students

FETCH REQUESTS

CONFIGURATIONS

Default

▼ Attributes

Attribute	^	Type
<div>S</div> name		String
+ —		

▼ Relationships

Relationship	^	Destination	Inverse
+ —			

▼ Fetched Properties

Fetch Property	^	Predicate
+ —		

Все данные сохраняются по ключу name.

1.7. IOS ПРИЛОЖЕНИЕ ПРОГНОЗА ПОГОДЫ

Цель:

Изучить пример разработки приложения Прогноз погоды

Получение данных о погоде через api:

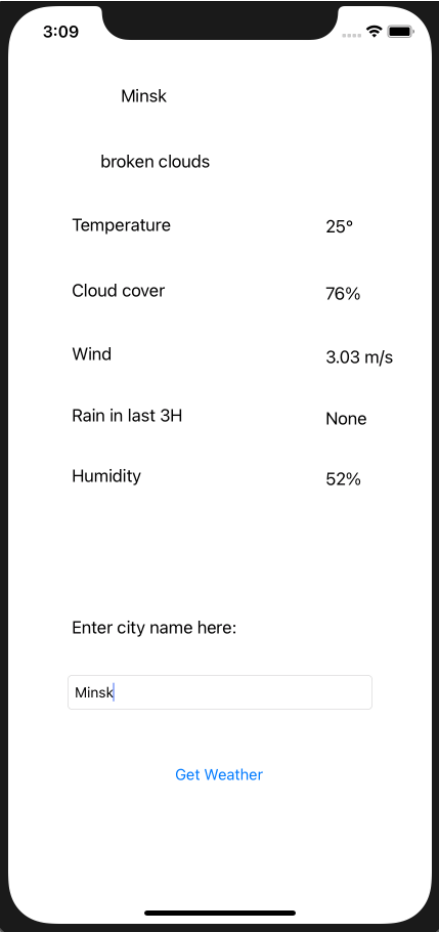
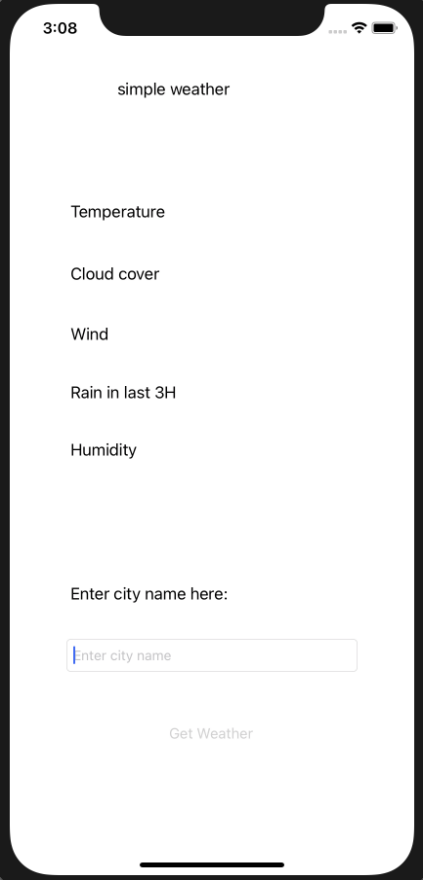
Raw data:
467 bytes

Human-readable data:
{
 "coord": {"lon": -82.4584, "lat": 27.9475}, "weather": [
 {"id": 701, "main": "Mist", "description": "mist", "icon": "50d"}]
 , "base": "stations", "main": {"temp": 295.86, "feels_like": 296
 .64, "temp_min": 293.75, "temp_max": 298
 .16, "pressure": 1013, "humidity": 94}, "visibility": 10000, "wind":
 {"speed": 2
 .57, "deg": 120}, "clouds": {"all": 20}, "dt": 1654600386, "sys":
 {"type": 2, "id": 2005199, "country": "US", "sunrise": 1654597991
 , "sunset": 1654647870}, "timezone": -14400, "id": 4174757, "name": "Tampa"
 , "cod": 200}

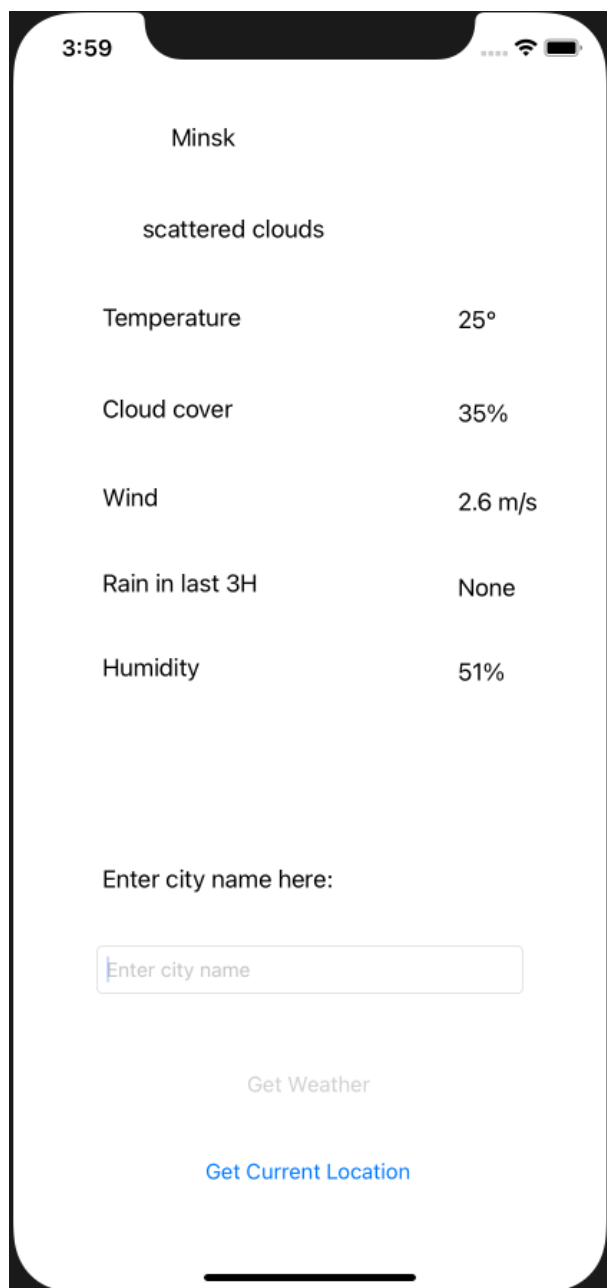
Перевод json в читабельный текст:

Date and time: 1654601182
City: Tampa
Longitude: -82.4584
Latitude: 27.9475
Temperature: 296.12
Humidity: 94
Pressure: 1013
Cloud cover: 20
Wind direction: 120 degrees
Wind speed: 2.57
Country: US
Sunrise: 1654597991
Sunset: 1654647870

Добавление пользовательского интерфейса:



Добавил кнопку получения погоды с текущей локации:



С кодом проще ознакомиться на гитхабе. Подводя итоги, можно сказать, что в ходе написания программы были разработаны следующие функции: получение погоды по городу, получение погоды по текущему местоположению. Вся погода получается в режиме реального времени через api сервиса OpenWeather.

2. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

2.1. СОЗДАНИЕ IOS ПРИЛОЖЕНИЯ С ХРАНЕНИЕ ДАННЫХ В NSUSERDEFAULTS И ФАЙЛЕ .PLIST

Создать iOS приложение на языке Swift с интерфейсом авторизации и хранением данных в UserDefaults, а так же выводом данных, хранимых в файле .plist, используя представление UICollectionView для вывода данных, согласно варианту задания.

Приложение должно поддерживать локализацию на 3 языка, два из которых русский и английский. Третьим языком можем быть белорусский, польский, французский, немецкий, итальянский и др.

Вариант задания:

Приложение «Парки
Минска» выводит список
парков Минска,
фотографию каждого.
При выборе парка
пользователь может
просмотреть сведения о
нем, время работы,
контакты и интересные
исторические факты

В приложение я добавил 3 парка - парк Челюскинцев, парк Победы и парк Горького.

Экран авторизации:

4:19

Login

Password

Sign In

Sign Up

Экран регистрации:

4:19

Login

E-Mail

Password

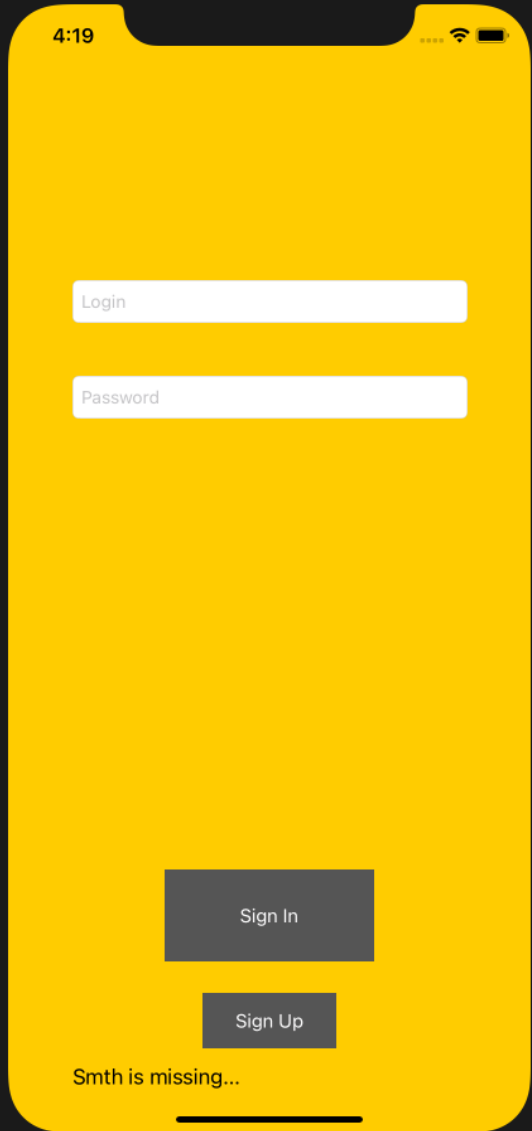
Confirm Password

Agree To Terms

Sign Up

Sign In

Ошибка при входе:



A mobile application login screen with a solid yellow background. At the top left, the time is 4:19. At the top right, there are icons for signal strength, Wi-Fi, and battery. Below the status bar, there are two white input fields: the first is labeled "Login" and the second is labeled "Password". Below these fields, there are two dark gray buttons: "Sign In" and "Sign Up". At the bottom of the screen, there is a text message "Smth is missing..." and a horizontal line representing the home indicator bar.

Вывод информации на разных языках:

Opened in 1945 in honor of the victory over Germany in the Great Patriotic War.

In 2008-2011, the park was reconstructed, sports grounds were renovated, a pedestrian bridge to Ptichy Island was built, and the bridge to Komsomolsky Island was reconstructed, where a real open-air gym is now equipped. The rescue station was rebuilt, embankments were equipped, several fountains were equipped, the motor ship Novy Nyasvizh began to run on the lake. A waterslide, a fort, a river lighthouse, umbrellas, cabins, showers and gazebos were installed on the renovated beach. Victory Park is a great place for a promenade, walking with children and spending a weekend. It is especially popular with hikers and cyclists, thanks to the convenient paths. There are more than 25 km of them on foot and almost 3 km on bicycles!

One of the main objects of the park is the Museum of the History of the Great Patriotic War, the new building of which was opened on July 2, 2014. The museum's funds contain authentic items of weapons and military equipment, household items, a large number of handwritten and documentary and photographic materials, fine art materials telling about the Great Patriotic War and the history of the struggle of the Belarusian people against the German



reconstruit, où une véritable salle de sport en plein air est désormais équipée. La station de sauvetage a été reconstruite, des remblais ont été équipés, plusieurs fontaines ont été équipées, le bateau à moteur Novy Nyasvizh a commencé à courir sur le lac. Un toboggan, un fort, un phare fluvial, des parasols, des cabines, des douches et des gazebos ont été installés sur la plage rénovée. Victory Park est un endroit idéal pour se promener, se promener avec des enfants et passer un week-end. Il est particulièrement apprécié des randonneurs et des cyclistes, grâce aux sentiers pratiques. Il y en a plus de 25 km à pied et près de 3 km à vélo !

L'un des principaux objets du parc est le musée de l'histoire de la Grande Guerre patriotique, dont le nouveau bâtiment a été inauguré le 2 juillet 2014. Les fonds du musée contiennent des armes et des équipements militaires authentiques, des articles ménagers, un grand nombre de documents manuscrits, documentaires et photographiques, du matériel d'art racontant la Grande Guerre patriotique et l'histoire de la lutte du peuple biélorusse contre les envahisseurs allemands.

Que regarder:
Musée de la Grande Guerre Patriotique ;
l'île aux oiseaux ;



4:21



Логин

Емэйл

Пароль

Подтвердите пароль

Согласен с услов...

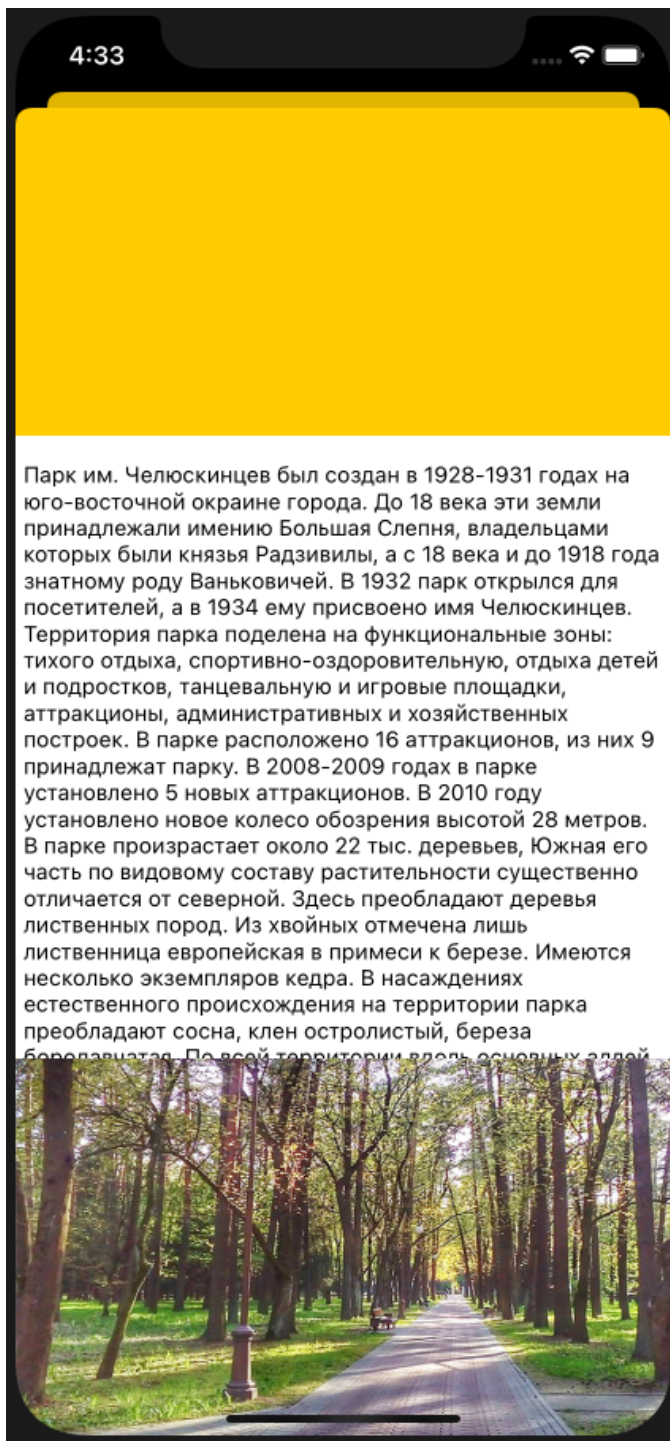


Зарегистрироваться

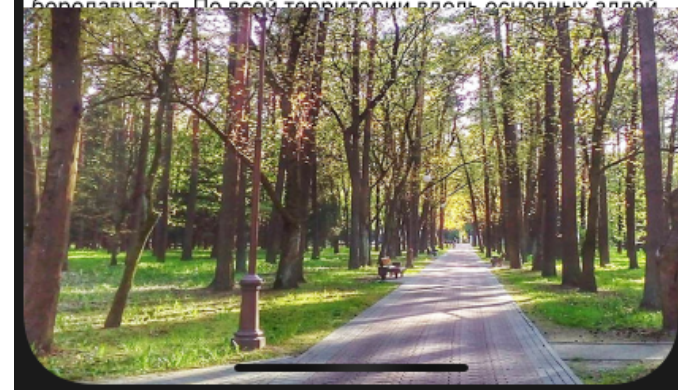
Войти



Как видно, локализация работает.
Полный экран с информацией о парках:



Парк им. Челюскинцев был создан в 1928-1931 годах на юго-восточной окраине города. До 18 века эти земли принадлежали имению Большая Слепня, владельцами которых были князья Радзивилы, а с 18 века и до 1918 года знатному роду Ваньковичей. В 1932 парк открылся для посетителей, а в 1934 ему присвоено имя Челюскинцев. Территория парка поделена на функциональные зоны: тихого отдыха, спортивно-оздоровительную, отдыха детей и подростков, танцевальную и игровые площадки, аттракционы, административных и хозяйственных построек. В парке расположено 16 аттракционов, из них 9 принадлежат парку. В 2008-2009 годах в парке установлено 5 новых аттракционов. В 2010 году установлено новое колесо обозрения высотой 28 метров. В парке произрастает около 22 тыс. деревьев, Южная его часть по видовому составу растительности существенно отличается от северной. Здесь преобладают деревья лиственных пород. Из хвойных отмечена лишь лиственница европейская в примеси к березе. Имеются несколько экземпляров кедра. В насаждениях естественного происхождения на территории парка преобладают сосна, клен остролистный, береза березавицкая. По всей территории вдоль основных аллей



По непонятной мне причине ячейки в Collection View не отображаются, однако если потыкать в левом верхнем углу желтой части экрана (тыкать в правом направлении), то парки переключаются.

2.2. СОЗДАНИЕ iOS ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ MAPKIT, CORELOCATION, COREDATA

Создать iOS приложение на языке Swift согласно варианту задания.

Для макета списка использовать UITableView. При создании приложения использовать библиотеки MapKit, CoreLocation, CoreData и сервис прогноза погоды согласно варианту задания.

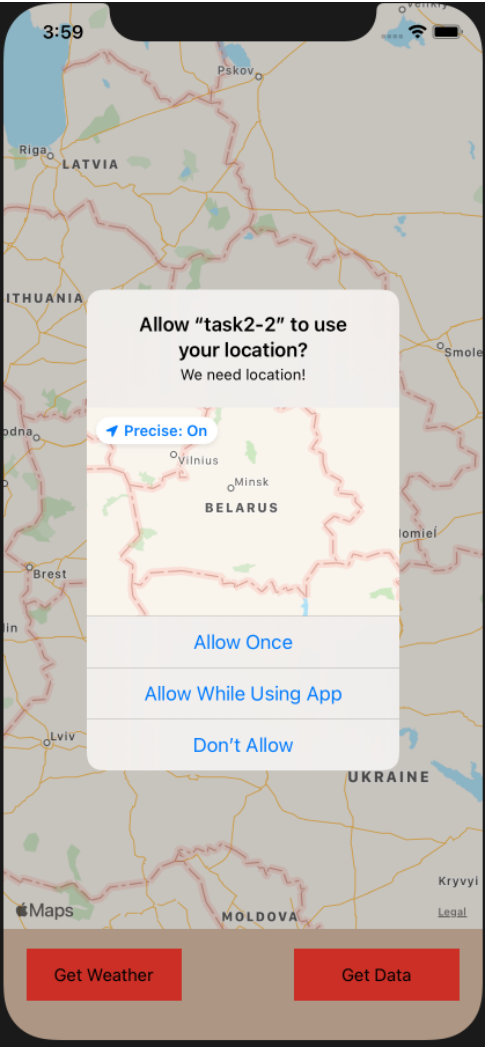
Приложение должно поддерживать локализацию на 3 языка, два из которых русский и английский. Третьим языком можем быть белорусский, польский, французский, немецкий, итальянский и др.

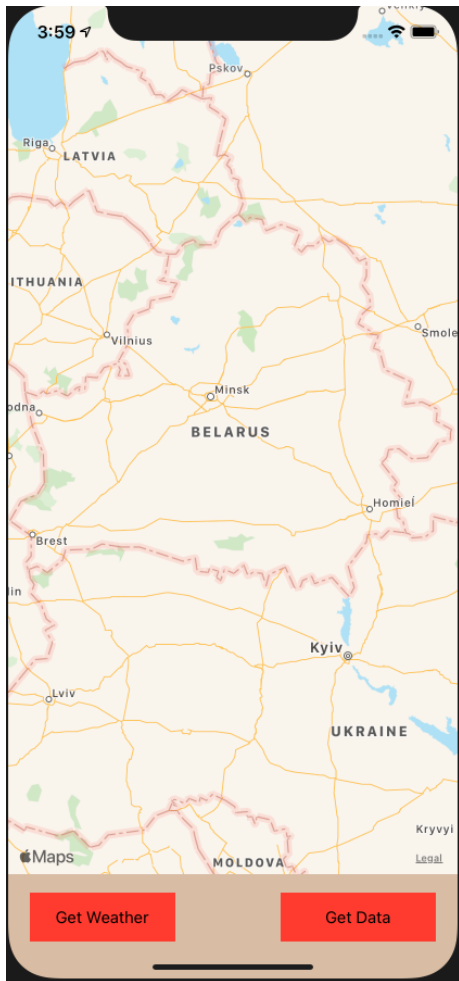
Код проекта распределить по группам (каталогам) согласно концепции MVC (Model, View, Controller) или MVVM (Model, View, ViewModel).

Вариант задания:

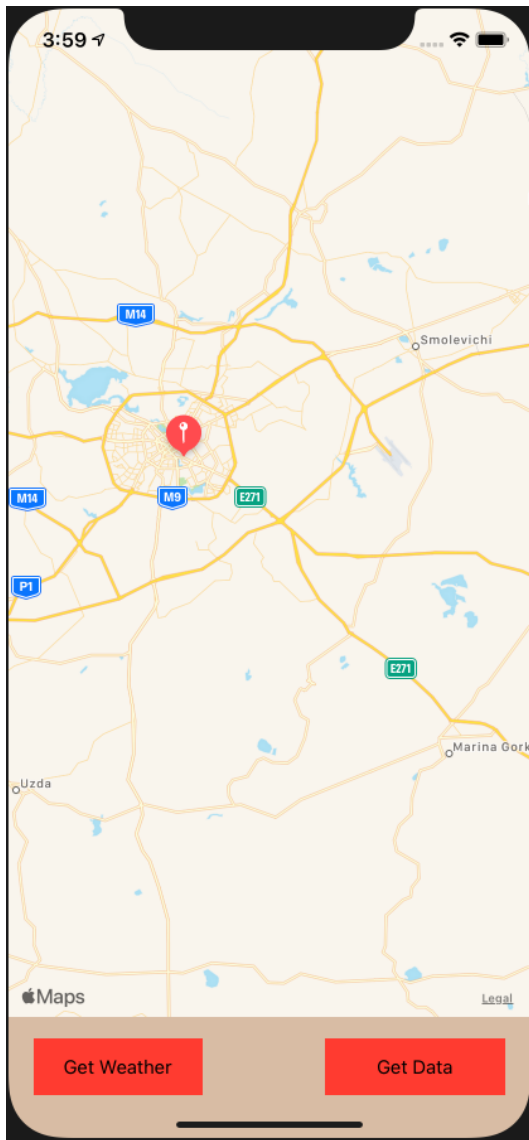
22	Приложение «Парки Минска» выводит список парков Минска, фотографию каждого. При выборе парка пользователь может просмотреть сведения о нем, время работы, контакты и интересные исторические факты	Требуется разработать программу интерактивную карту Могилевской области, используя CoreData, CoreLocation и MapKit. Пользователь выбирает город на карте, и ему предлагается список историческая справка о городе, а так же как доехать в выбранный город из Минска. При выборе города на карте пользователь может посмотреть прогноз погоды получая данные от сервиса https://openweathermap.org/api . Данные должны храниться в базе данных sqlite, доступ через CoreData.
----	--	---

Главный экран приложения:

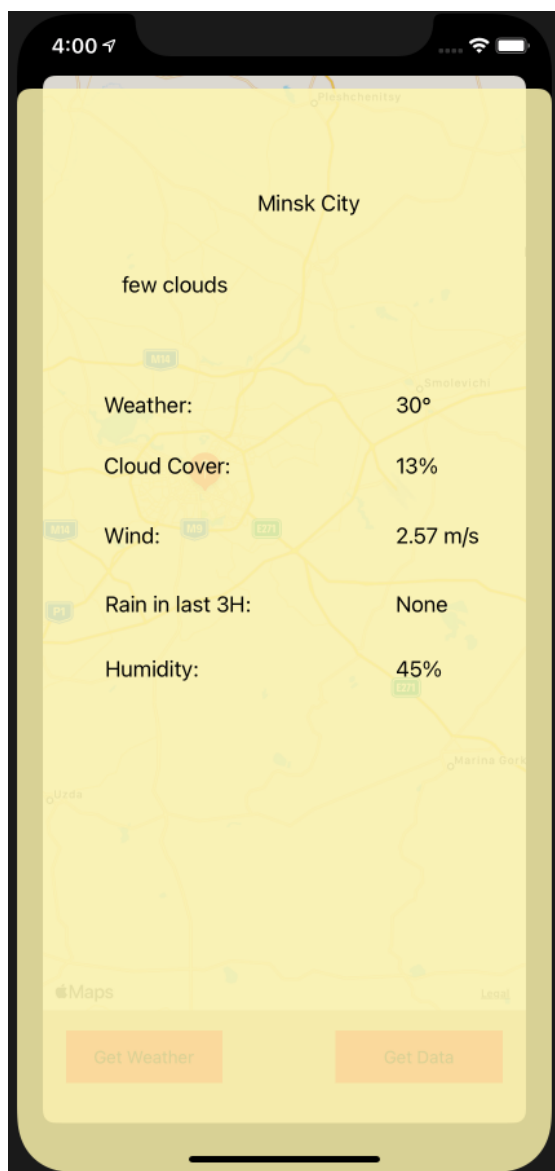




Выбор определенного города:



Показ погоды:



Показ данных о городе (не доделан:():

